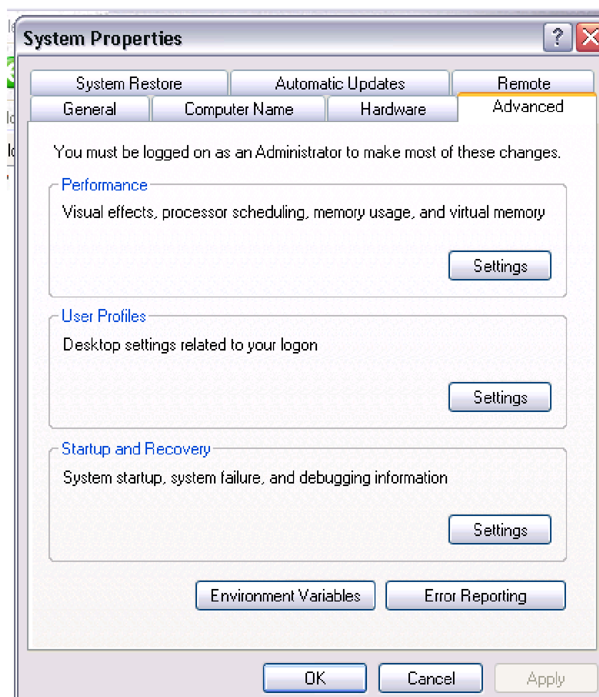
Modul-1

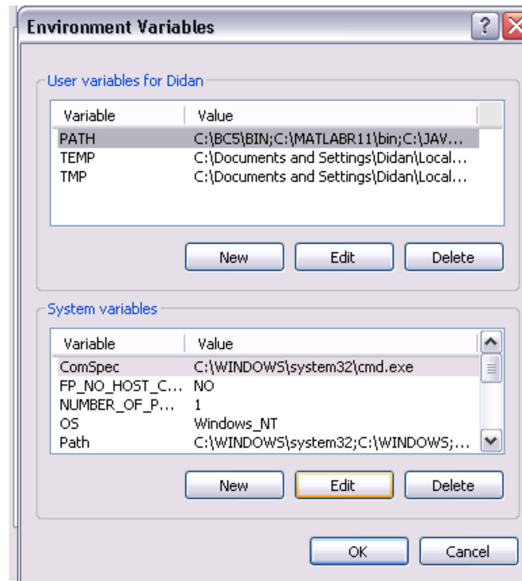
Judul : **Instalasi Java**

Prosedur Pelaksanaan :

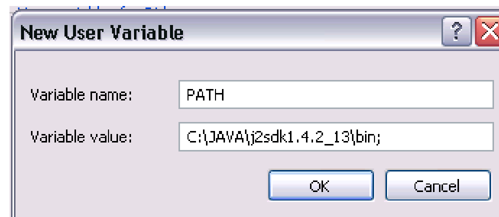
1. Siapkan JDK yang telah disediakan, lalu installah seperti biasa, usahakan agar direktori tempat JDK ini mudah diakses, misalkan kita letakkan pada c:\ Java
2. Setelah terinstal lakukan setting classpath jika anda ingin menggunakan editor text seperti notepad, fungsi dari setting classpath ini agar program java yang anda buat dapat dijalankan dimana saja di seluruh partisi hardisk. Cara setting class path ini sebagai berikut
   a. Cari direktori bin yang ada terdapat pada direktori tempat Java anda terinstall
   b. Lalu catat alamat tempat diletakkan nya folder bin tersebut
   c. Setelah itu klik kanan pada MyComputer pilih properties, lalu klik pada tab advance



   d. Setelah itu pilih Environtment Variabel



e. Lalu pada variable PATH pilih edit karena kita akan menambahkan classpath yang baru, maka akan muncul jendela berikut



f. Tambahkan alamat terletaknya folder bin dalam hasil instalasi java.pada kolom Variable Value. Setelah itu tekan OK.

3. Setelah di setting classpath restart komputer anda agar terjadi perubahan yang diinginkan
4. untuk mengecek apakah JDK yang anda install sudah terpasang dengan baik adalah dengan menggunakan command prompt ketikan C:\> java, atau C:\> javac, jika melihat tulisan sebagai berikut itu berarti JDK anda sudah terinstall dengan baik

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

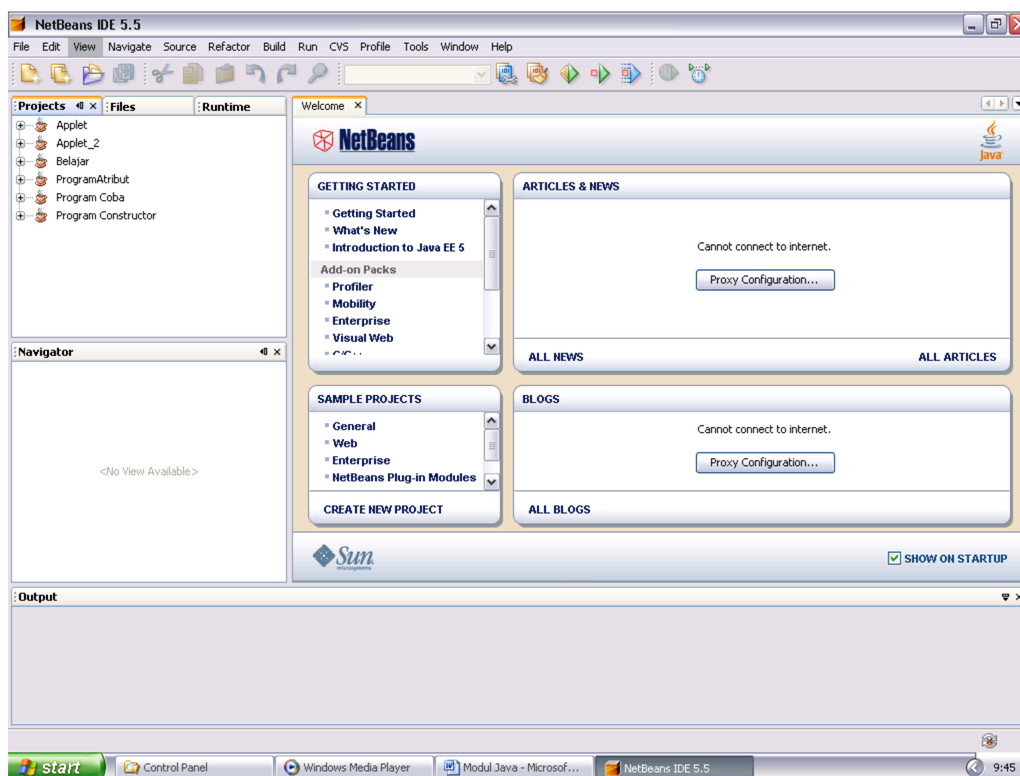
C:\Documents and Settings\Didan>java
Usage: java [-options] class [args...]
           (to execute a class)
   or  java [-options] -jar jarfile [args...]
           (to execute a jar file)

where options include:
    -client       to select the "client" VM
    -server       to select the "server" VM
    -hotspot      is a synonym for the "client" VM  [deprecated]
                  The default VM is client.

    -cp <class search path of directories and zip/jar files>
    -classpath <class search path of directories and zip/jar files>
                  A ; separated list of directories, JAR archives,
                  and ZIP archives to search for class files.
    -D<name>=<value>
                  set a system property
    -verbose[:class|gc|jni]
                  enable verbose output
    -version      print product version and exit
    -version:<value>
```

5. sedangkan jika kita menggunakan IDE (Integrated Development Environment) kita tinggal menginstall nya setelah JDK terinstall, ada beberapa IDE yang bisa di pakai antara lain JBuilder buatan Borland, Eclipse, NetBeans, dll

contoh tampilan IDE NetBeans



Latihan 1. **Hello World**
Perhatikan listing code berikut

```
public class Hallo
{
   public static void main(String[] args)
   {
      System.out.println("Apa Kabar UMC ");
   }
}

Public class haloName
{
     public static void main(String[] args)
     {
        System.out.print("Nama ");
        System.out.print("Saya");
        System.out.print("Joni "+"Iskandar")
     }
}
```

Modul-2

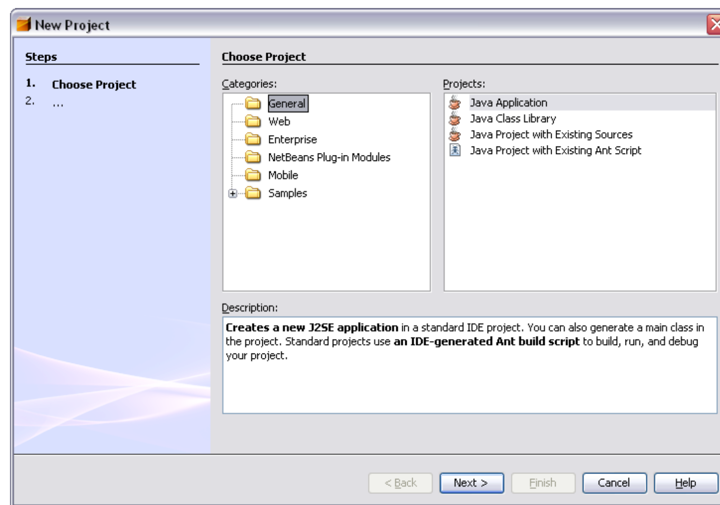
Judul : **Program Java Pertama, Varibel, Tipe Data**

Prosedur Pelaksanaan :

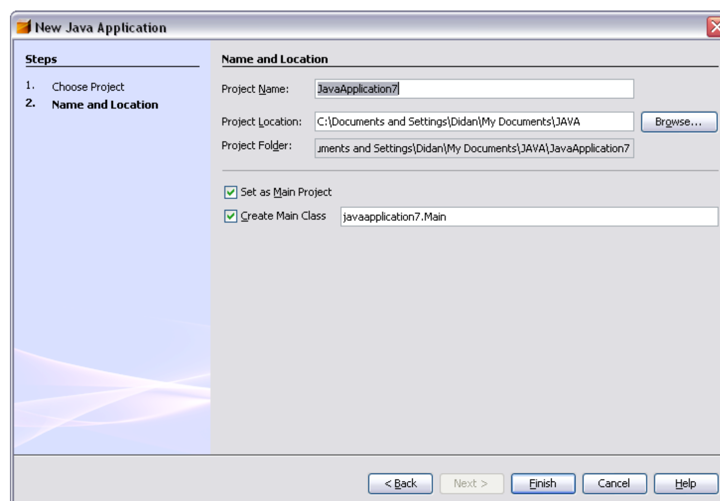
Latihan 1. **Program Java Pertama**

Menggunakan IDE Netbeans

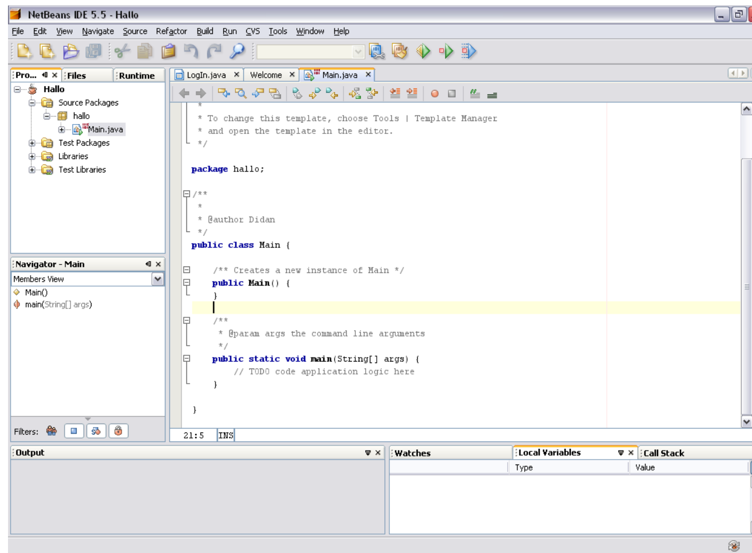
1. Buka NetBeans yang telah terisntall, maka akan muncul seperti berikut
2. Lalu pilih File → New Project, maka akan muncul jendela sebagai berikut



3. karena kita akan membuat program java " biasa" maka kita pada kolom Categories kita pilih General dan pada kolom Projects kita pilih Java Application, setelah itu kita pilih next



4. beri nama project yang akan kita buat, lalu pilih lokasi dimana kita akan menempatkan lokasi project yang kita buat, setelah itu pilih finish
5. maka akan muncul sebuah kotak editor yang berisi nama class dari java

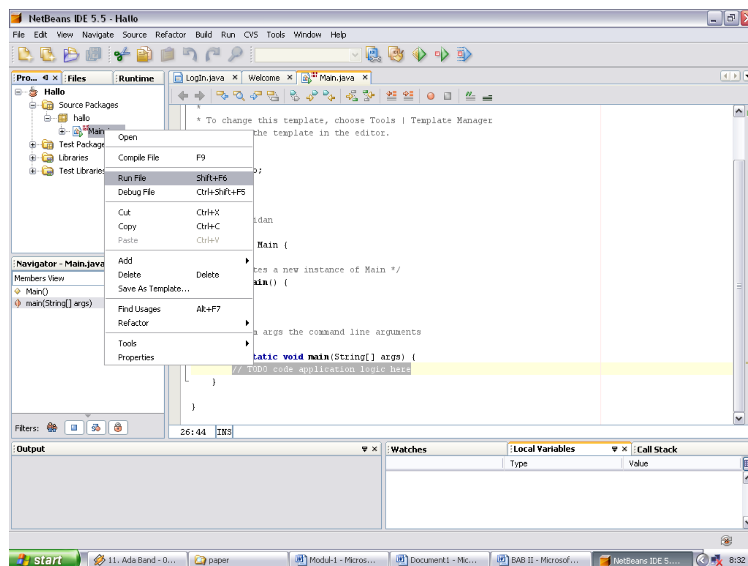


6. lalu kita ketikkan kode berikut di bawah tulisan // TODO code application logic here

```
System.out.println(" Hello UMC ");
```

Kode ini akan menampilkan kata Hello UMC

7. untuk menjalankan nya kita klik kanan pada nama program dan pilih Run File



8. Hasil dari program tadi ditampilkan pada jendela output yang ada pada kiri bawah editor

Menggunakan Command Line

9. Ketikan source code java pada text editor. Misal pada notepad, notepad++
10. Simpan nama file sama dengan nama class yang anda buat (case sensitive) dan berekstensi java
11. Buka command prompt pada windows lalu arahkan menuju direktori dimana anda meletakkan file java tersebut.
12. Pada command prompt ketikkan A:/>javac <nama File>.java

13. Jika sukses maka tidak akan muncul pesan kesalahan
14. Hasil compilasi tadi menghasilkan file dengan ekstensi .class
15. Jalankan file hasil compilasi tadi dengan cara mengetikkan java <nama File yang telah dikompilasi>
16. Jika pada proses compilasi maupun running tadi tidak dapat dijalankan, periksa kembali setting classpath anda.

Listing 2 **Variabel**

Perhatikan listing kode di bawah ini

```
public class ContohVariable
{
   public static void main( String[] args )
   {
      int K = 10;
      char x;
      x = ‘A’;
      double hasil1;
      int hasil2;

      int a = 5;
      int b;

      b = 6;
      // K = 25;
      hasil1 =  a + b / K;
      hasil2 = a * b;

      System.out.println( “ Nilai K adalah “ + K );
      System.out.println( “Nilai dari x= ” + x );

      System.out.println(a +” + “+ b +” / “+K +” = “+hasil1);
      System.out.println(a +” * “+ b +” = “+hasil2);

   }
}
```

Listing 3 **Tipe Data**

Perhatikan listing kode di bawah ini

```
public class TipeData
{
        public static void main(String[] args)
    {
            int number = 10;
            char letter = 'a';
            boolean result = true;
            String str = "Hello";

            System.out.println("Number = " + number);
            System.out.println("letter = " + letter);
            System.out.println("Result = " + result);
```

```
            System.out.println("str = " + str);
        }
}
```

Latihan 4 **Operator Aritmatika, Variabel**

Perhatikan Listing Kode di bawah ini

```
public class aritmatikaDemo
{
   public static void main(String[] args)
   {
      //sedikit angka
      int i = 37;
      int j = 42;
      double x = 27.475;
      double y = 7.22;

      System.out.println("Nilai Variable...");
      System.out.println(" i = " + i);
      System.out.println(" j = " + j);
      System.out.println(" x = " + x);
      System.out.println(" y = " + y);

      //penjumlahan angka
      System.out.println("Penjumlahan...");
      System.out.println(" i + j = " + (i + j));
      System.out.println(" x + y = " + (x + y));

      //pengurangan angka
      System.out.println("Pengurangan...");
      System.out.println(" i - j = " + (i - j));
      System.out.println(" x - y = " + (x - y));

      //perkalian angka
      System.out.println("Perkalian...");
      System.out.println(" i * j = " + (i * j));
      System.out.println(" x * y = " + (x * y));

      //pembagian angka
      System.out.println("Pembagian...");
      System.out.println(" i / j = " + (i / j));
      System.out.println(" x / y = " + (x / y));

      //menghitung hasil modulus dari pembagian
      System.out.println("Modulus...");
      System.out.println(" i % j = " + (i % j));
      System.out.println(" x % y = " + (x % y));

      //tipe penggabungan
      System.out.println("Operator Gabungan...");
      System.out.println(" j + y = " + (j + y));
      System.out.println(" i * x = " + (i * x));
   }
}
```

Latihan 5 **Operator ? : (ternary)**
Perhatikan Listing Kode di bawah ini

```
public class KondisiOperator
{
  public static void main( String[] args )
  {

     String    status = "";
     int  grade = 80;

     //mendapatkan status pelajar
     status = (grade >= 60)?"Passed":"Fail";

    //print status
     System.out.println( status );
  }
}
```

Modul-3

Judul : **Input Form Keyboard**, **Strukutur Kontrol**

Prosedur Pelaksanaan :

1. <u>Statement if – then – else</u>

Perhatikan Listing Kode berikut

```
public class inputAngka
{
    public static void main(String[] args)
    {
        String Angka ;
        BufferedReader bfr = new BufferedReader(new
     InputStreamReader(System.in));
        try
        {
            System.out.println("Masukkan Angka 1 - 5 : ");
            Angka = bfr.readLine();
            int i = Integer.parseInt(Angka);
            if(i == 1)
            {
                System.out.println("Anda Memasukkan Angka 1 ");
            }
            else if(i == 2)
            {
                System.out.println("Anda Memasukkan Angka 2 ");
            }
            else if(i == 3)
            {
                System.out.println("Anda Memasukkan Angka 3 ");
            }
            else if(i == 4)
            {
                System.out.println("Anda Memasukkan Angka 4 ");
            }
            else if(i == 5)
            {
                System.out.println("Anda Memasukkan Angka 5 ");
            }
            else
            {
                System.out.println("Angka  yang  anda  masukkan  tidak  sesuai
");
            }

        }
        catch(IOException e)
        {
            System.err.println("Error");
        }
    }
}
```

2. <u>Statement While</u>

   Perhatikan Listing Kode berikut

```
public class cetakWhile
{
    public static void main(String[] args)
   {
        int i = 0;
        while(i <= 100)
        {
            System.out.println("Hallo UMC");
            i++;
        }
    }
 }
```

   Listing kode diatas akan menampilkan tulisan Hallo UMC sebanyak 100 kali

3. <u>Statement do – while</u>

   Perhatikan Listing Kode berikut

```
public class cetakDoWhile
{
    public static void main(String[] args)
   {
        int x = 0;
        do
        {
            System.out.println("Selamat Pagi");
            x++;
        }while(x < 100);
    }
}
```

4. <u>Statement for</u>

   Perhatikan Listing Kode berikut

```
public class cetakFor
{

    public static void main(String[] args)
    {
        for(int x = 0; x < 100; x++)
        {
            System.out.println("Apa Kabar ?");
        }
    }
 }
```

5. <u>Statement switch - case</u>

   Perhatikan Listing Kode berikut

```
import java.io.*;
```

```
public class inputSwitch
{
 public static void main(String[] args)
 {

      String Angka;
    BufferedReader       bfr       =       new       BufferedReader(new
    InputStreamReader(System.in));
       System.out.println("Masukkan Angka : ");
        try
        {
             Angka = bfr.readLine();
             int i = Integer.parseInt(Angka);
             switch(i)
         {
             case 5:
                 System.out.println("Anda Memasukkan Angka 5 ");
                 break;
             case 4:
                 System.out.println("Anda Memasukkan Angka 4 ");
                 break;
             case 3:
                 System.out.println("Anda Memasukkan Angka 3 ");
                 break;
             case 2:
                 System.out.println("Anda Memasukkan Angka 2 ");
                 break;
             case 1:
                 System.out.println("Anda Memasukkan Angka 1 ");
                 break;
             default:
                 System.out.println("Anda Memasukkan angka yang salah ");
         }
        }
        catch (IOException ex)
        {
            ex.printStackTrace();
        }

    }
}
```

Berikut ini adalah langkah-langkah yang diperlukan untuk mendapatkan input dari keyboard:

1. Tambahkan di bagian paling atas code Anda:

   ```
   import java.io.*;
   ```

2. Tambahkan statement berikut:

   ```
   BufferedReader dataIn = new BufferedReader(new InputStreamReader( System.in)

   );
   ```

3. Deklarasikan variabel String temporary untuk mendapatkan input, dan gunakan fungsi readLine() untuk mendapatkan input dari keyboard. Anda harus mengetikkannya di dalam blok try-catch:

   ```
   Try
   ```

```
{
     String temp = dataIn.readLine();
}
catch( IOException e )
{  System.out.println("Error in getting input");   }
```

4. Berikut Source Code Lengkapnya

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
public class GetInputFromKeyboard
{
    public static void main( String[] args )
    {
        BufferedReader dataIn = new BufferedReader(new InputStreamReader(
        System.in) );
        String name = "";
        System.out.print("Please Enter Your Name:");
        Try
        {
            name = dataIn.readLine();
        }
        catch( IOException e )
        {
              System.out.println("Error!");
        }
        System.out.println("Hello " + name +"!");
    }
}
```

Meng-casting tipe data yang berbeda
Casting tipe data ini digunakan untuk merubah tipe data
Misal : tipe data string menjadi integer

```
String angka;
int a;
a = Integer.parseInt(angka);
```

Modul-4

Judul : **Java Array**

Teori **:** Array adalah sebuah variabel yang dapat menampung beberapa tipe data yang sama dengan nilai variabel yang berbeda. Sebuah array akan menyimpan beberapa item data yang memiliki tipe data sama didalam sebuah blok memori yang berdekatan yang kemudian dibagai menjadi beberapa ruang. Array adalah sebuah variabel/sebuah lokasi tertentu yang memiliki satu nama sebagai identifier, namun identifier ini dapat menyimpan lebih dari sebuah nilai.

Pendeklarasian Array

Pada java ada 2 cara untuk mendeklarasikan sebuah array :

1. Anda harus menetukan tipe data yang akan digunakan, lalu dikuti tanda kurung siku [], lalu diikuti dengan identifiernya.

```
int [] nomor;
```

2. Pada cara yang kedua tanda siku dapat diletakkan setelah identifiernya.

```
int nomor[];
```

setelah anda mendeklarasikan sebuah array anda diharuskan untuk men-create array tersebut dan menentukan berapa panjang array tersebut dengan konstruktor.

```
// deklarasi
int[] nomor;
// instance
nomor = new int[10];
```

atau dapat juga ditulis sebagai berikut

```
int nomor[] = new int[10];
String []nama = new String[3];
```

<u>Latihan 1</u> **Membuat Array**

```
String hari[] = {"Senin","Selasa","Rabu","Kamis","Jum'at","Sabtu","Minggu"};
int i = 0;
while(i < hari.length)
{
      System.out.println(hari[i]);
       i++;
}
```

Latihan 2 **Array Multidimensi**

```
int no[][] = {{1,2},{3,4},{5,6}};
for(int i = 0; i < no.length;i++)
{
            System.out.println("No : "+no[i][0]);
            System.out.println("No : "+no[i][1]);
}

String data[][] = {{"Ujang ","UMC"},{"Udin","CIC"},{"Dian","STIKOM"}};
for(int i = 0; i < data.length;i++)
{
            System.out.println("Nama   : "+data[i][0]);
            System.out.println("Kampus : "+data[i][1]+"\n");
}
```

Latihan 3 **Mengcopy Array**

Untuk mengcopy array java menyediakan method sendiri, yaitu dengan menggunakan method arraycopy
Adapun format penulisannya sebagai berikut :
System.arraycopy(arraySumber, indexArraySumber, arrayTujuan, indexArrayTujuan, panjangDataYangInginDicopy)

```
int[] dimensi1 = {2,5,3,6,8,1,4,6,9,10,7,5};
int[] x = new int[5];

System.arraycopy(dimensi1, 0, x, 0, 5);
for(int i = 0; i < x.length; i++)
{
           System.out.println("x["+i+"] : "+x[i]);
}
```

Latihan 4. **Menjumlahkan array**

Untuk menjumlahkan array diperlukan 2 buah array yang memiliki ordo yang sama begitu juga untuk pengurangan array

```
int a[][] = {{2,3,4},{4,5,6},{7,8,9}};
int b[][] = {{1,1,1},{2,2,2},{3,3,3}};
int c[][] = new int[3][3];

for(int i = 0; i < 3; i++)
{
      for(int j = 0; j < 3; j++)
      {
            c[i][j] = 3 * b[i][j];
      }
}
```

Modul-5

Judul : **Membuat Class Sederhana**

Teori : Seperti kita ketahui bahwa Java adalah bahasa yang murni 100% Object-Oriented. Sehinggga mau tidak mau kita juga harus belajar tentang object-oriented programming, dan kita juga harus belajar berfikir objek untuk mudah memahami bahasa Java.

Berbeda dengan teknik pemrograman secara struktural. Object-Oriented Programming adalah suatu cara dalam memprogram, dimana kita melihat suatu program sebagai suatu benda hidup, sehingga sebuah program komputer tidak bersifat pasif, tetapi ia akan terlihat 'hidup'. Hal ini didasarkan pada paradigma dalam dunia nyata. Yaitu memandang data seperti memandang benda-benda dalam kehidupan nyata, memiliki fleksibilitas dalam pendefinisian data, dll.

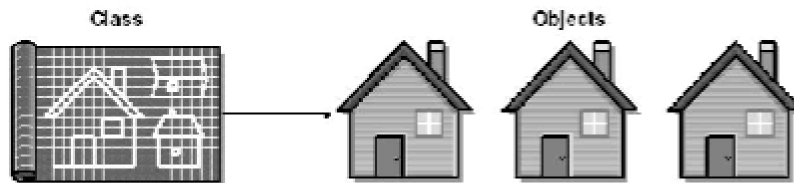
Ada tiga karakteristik utama dari sebuah pemrograman yang berorientasi Object, yaitu :

- Encapsulation
- Inheritance
- Polymorphism

Sebelum kita masuk ke karakteristik OOP kita akan bahas sedikit beberapa konsep dasar dalam OOP

- **Class**

Class adalah blueprint atau prototype dari objek-objek. kita misalkan sebagai berikut :



Class adalah blueprint / rancang bangun yang nantinya akan dibuat sebuah / beberapa object. Class mengizinkan anda dalam mendeklarasikan tipe data baru

Contoh lain dari analogi sebuah class dengan object adalah sebagai berikut :

Misalkan kucing dari classs karnivora, lumba-lumba dari class mamalia

Dalam Java kita tidak dapat mendefinisikan objek secara langsung, melainkan kita bekerja dengan kelas dari sebuah objek.

Sebuah class dalam Java dideklarasikan sebagai berikut :

```
[atribut_akses] class [Identifier] [super] [interface] [body]
```

Sebuah class harus mempunyai kata kunci **class**. Kata kunci ini dapat didahului oleh atribut_akses yang bersifat opsional. Kata kunci class harus diikuti nama class berupa identifier. Baru kemudian

dapat diikuti dengan nama-nama dari superclass dan/ atau super interface. Baru kemudian diikuti dengan block { } yang merupakan body dari class.

Contoh nama dari sebuah class :

```
public class Persegi
{
    private int panjang;
    private int lebar;
    public Persegi()
    {
    }
}
```

" ingat nama class harus sama dengan nama file "

Atribut_akses adalah salah satu atau gabungan dari kata kunci berikut :

```
public
protected
private
abstrack
static
final
```

- **Access Specifier**

Access Spesifier Untuk Variable :

Jenis variable	Lingkup pengaksesan	Daur hidup
Variabel instant	Tunduk terhadap kondisi berikut : 1. private – hanya metode dalam class yang dapatm mengakses 2. public – sembarang class dapat mengakses. 3. friendly – kelas atau sembarang class dalam paket 4. protected – sembarang subclass dari class dapat mengakses	Saat instance diciptakan hingga tidak ada lagi referensi terhadap instance.
Variabel class	Tunduk terhadap lingkup yang sama dengan variable instance	Saat kelas dimuat hingga tidak ada lagi referensi terhadap class tersebut.
Variabel lokal	Kode didalam blok yang sama	Saat kode blok masih aktif.

Access Specifier untuk method :

Access Specifier	Lingkup pengaksesan	Pemakaian
Public	Metode dapat diakses sembarang class	Antarmuka eksternal class anda
Private	Metode dapat diakses oleh metode dalam class yang sama	Metode untuk pemakaian internal class (tidak

		berhubungan dengan pamakai class)
Tidak secara eksplicit disebutkan (friendly)	Metode dapat diakses oleh metode didalam class atau metode dalam class lain yang berada pada paket yang sama.	Metode yang anda inginkan dapat diakses oleh kelas yang berkaitan
Protected	Metode dapat diakses oleh metode dalam subclass dari class tersebut	Metode yang ingin diakses oleh subclass dari class tersebut.

- **Object**

Objek, adalah entiti yang memiliki keadaan, Behaviour dan identitas yang tugasnya dirumuskan dalam suatu lingkup masalah dengan naik. Inilah instance yang sebenarnya dari sebuah class. Ini juga dikenal sebagai instance.

Object adalah representasi nyata dari sebuah class.

Kita dapat menyatakan sesuatu yang ada di sekitar kita sebagai suatu objek, dari yang nyata, seperti Komputer, pohon, hewan, bangunan, dll. Hingga yang abstrak, seperti hutang, penjualan barang, dll.

Dalam Java kita tidak dapat mendefinisikan objek secara langsung, melainkan kita bekerja dengan kelas dari sebuah objek. kita dapat meng-instance sebuah class menjadi object dengan kata kunci **new.**

Contoh :

```
public class hitungPersegi
  {
   public static void main(String[] args) {
   // membuat objek
   Persegi p = new persegi() ;
   }
  }
```

Perhatikan perbedaan antara class dan obyek yang dicontohkan dalam dunia nyata

Class Motor		Object Motor A	Object Motor B
Variabel Instance	Nomor Plat	E 1234 GB	E 4569 GH
	Warna	Hitam	Merah
	Pabrikan	Yamaha	Honda
	Kecepatan	40 km/ jam	50 km/jam
	Method Akslelerasi		

Method	Method Rem
Instance	Method standing up

- **Field**

  menyatakan informasi yang dibawa kelas dan menjadi parameter dari objek dari kelas yang bersangkutan. Misal ada kelas bangun geometri kubus, maka parameter bagi objek kubus itu adalah panjang. Maka kita akan menulis field itu dalam java dengan format sebagai berikut :

```
Atribut_akses tipe_data Nama_field;
```

  Maka kita tulis sebagai berikut :

```
private int panjang;
```

  Dalam Java, istilah variable dipakai untuk menyatakan field, local-variable, parameter, dan komponen array.

Istilah field sendiri dipakai untuk menyatakan variable yang merupakan bagian langsung dari sebuah class atau interface

- **Method**

  Sebuah method menjelaskan behaviour dari sebuah object. Method juga dikenal sebagai fungsi atau prosedur.

  Secara sederhana method dapat dikatakan sebagai operasi yang dapat dilakukan oleh object dari class. Operasi ini bisa berupa menerima dan menampilkan data, memanipulasi field, atau mempengaruhi objek lain.

  Berikut ini adalah format penulisan dalam mendeklarasikan method dalam body kelas.

```
Atribut_akses tipe_kembalian Nama_Method(arg1,arg2,...argn)
{
      // body of method
}
```

  Berikut ini adalah karakteristik dari sebuah method

  1. Dapat mengembalikan satu nilai atau tidak sama sekali
  2. Dapat diterima beberapa parameter yang dibutuhkan atau tidak ada parameter sama sekali. Parameter bisa juga disebut sebagai argumen dari fungsi
  3. Setelah method telah selesai dieksekusi, dia akan kembali pada method yang memanggilnya

1. Membuat Class mahasiswa

```
public class mahasiswa
{
      private int nim;
      private String nama;
      prvate String alamat;
      private String email;
```

```
        public String getNama()
        {
             return nama;
        }
        public int getNim()
        {
            return nim;
        }
        public void setNama(String nama)
        {
            this.nama = nama;
        }
        public void setNim(int nim)
        {
            this.nim = nim;
        }
}
```

2. Membuat Class mainMahasiswa

```
public class mainMahasiswa
{
    public static void main(String[] args)
    {
        Mahasiswa mhs1 = new Mahasiswa();
        Mahasiswa mhs2 = new Mahasiswa();

        mhs1.setNama("Joni");
        mhs1.setNim("00123456");
        mhs1.setEmail("Joni@yahoo.com");
        mhs1.setAlamat("Cirebon");

        mhs2.setNim("00123457");
        mhs2.setAlamat("Kutaraja");
        mhs2.setEmail("ayu@ayu.com");
        mhs2.setNim("ayu");

        //Menampilkan Data
        System.out.println("Nama  : "+mhs1.getNama());
        System.out.println("NIM   : "+mhs1.getNim());
        System.out.println("Email : "+mhs1.getEmail());
        System.out.println("Alamat: "+mhs1.getAlamat());

    }
}
```

3. Membuat Class Persegi Panjang

```
public class PersegiPanjang
{
    private int panjang;
    private int lebar;

    public PersegiPanjang(int panjang, int lebar)
    {
        this.panjang = panjang;
        this.lebar = lebar;
    }

    public int getLebar()
    {
        return lebar;
    }

    public void setLebar(int lebar)
    {
        this.lebar = lebar;
    }

    public int getPanjang()
```

```
    {
        return panjang;
    }

    public void setPanjang(int panjang)
    {
        this.panjang = panjang;
    }

    public int Luas()
    {
        int Luas;
        Luas = panjang * lebar;
        return Luas;
    }
}
```

4. Membuat Class mainPersegiPanjang

```
public class mainPersegiPanjang
{
    public static void main(String[] args)
    {
        PersegiPanjang kotak =  new PersegiPanjang();

        kotak.setLebar(10);
        kotak.setPanjang(15);

        //Menampilkan Luas Persegi Panjang
        System.out.println("Luas : "+kotak.getLuas());
    }
}
```

Modul-6

Judul : **Membuat Class (Overloading Constructor, Overloading Method)**

Teori :
Overloading Constructor adalah Membuat 2 Constructor atau lebih yang memiliki beberapa parameter dalam sebuah class, seperti kita tahu bahwa construtor digunakan sebagai inisialisasi sebuah class, dengan adanya overloading constructor kita dapat membuat beberapa objek yang serupa tapi tak sama
Contoh :
Pada class Mahasiswa yang kemarin telah kita buat kita tidak melihat adanya construtor, akan tetapi pada saat kita meng-instance objek mahasiswa kita mendeklarasikan `new Mahasiswa();` hal itu terjadi karena java secara otomatis membuat sendiri construtornya secara default.

Lat 1. Class Mahasiswa

```
public class mahasiswa
{
    private String nim;
    private String nama;
    private String alamat;
    private String email;
    private static int jumlah;

    public Mahasiswa()
    {
        jumlah++;
    }
    public Mahasiswa(String nim)
    {
        this.nim = nim;
        jumlah++;
    }
    public String getAlamat()
    {
        return alamat;
    }

    public void setAlamat(String alamat)
    {
        this.alamat = alamat;
    }

    public String getEmail()
    {
        return email;
    }

    public void setEmail(String email)
    {
        this.email = email;
    }

    public String getNama()
    {
        return nama;
    }
```

```
    public void setNama(String nama)
    {
        this.nama = nama;
    }

    public String getNim()
    {
        return nim;
    }

    public void setNim(String nim)
    {
        this.nim = nim;
    }

    public static int getJumlah()
    {
        return jumlah;
    }
    public void getData()
    {
        System.out.println("Nama   : "+nama);
        System.out.println("Nim    : "+nim);
        System.out.println("Email  : "+email);
        System.out.println("Alamat : "+alamat);
    }
    public void getData(int NilaiUts, int NilaiUas)
    {
        System.out.println("Nama        : "+nama);
        System.out.println("Nim         : "+nim);
        System.out.println("Nilai UTS   : "+NilaiUts);
        System.out.println("Nilai UAS   : "+NilaiUas);
        System.out.println("Nilai Akhir : "+((0.4 * NilaiUts) + (0.6 *
NilaiUas)));
    }
}
```

```
public class mainMahasiswa
{
    public static void main(String[] args)
    {
        Mahasiswa mhs1 = new Mahasiswa();
        Mahasiswa mhs2 = new Mahasiswa();
        Mahasiswa mhs3 = new Mahasiswa("0012399");

        mhs1.setNama("Joni");
        mhs1.setNim("00123456");
        mhs1.setEmail("Joni@yahoo.com");
        mhs1.setAlamat("Cirebon");

        mhs2.setNim("00123457");
        mhs2.setAlamat("Kutaraja");
        mhs2.setEmail("ayu@ayu.com");
        mhs2.setNim("ayu");

        //Menampilkan Data
        System.out.println("Nama  : "+mhs3.getNama());
        System.out.println("NIM   : "+mhs3.getNim());
        System.out.println("Email : "+mhs3.getEmail());
        System.out.println("Alamat: "+mhs3.getAlamat());

        //Menampilkan Jumlah Data
        System.out.println("Jumlah Mahasiswa : "+Mahasiswa.getJumlah());

        //Menampilkan Data Mhs1
        mhs1.getData();
        System.out.println("");

        //Menampilkan Data Mhs2
        mhs2.getData(70, 90);
    }
}
```

Modul-7

Judul : **Inheritance (Penurunan Sifat / Pewarisan) & Polymorfisme**

Teori : Inheritance dan Polymorfisme adalah kedua sifat yang dimiliki dalam Konsep OOP. Dalam inheritance, method overriding berbeda dengan method overloading. Kalau method overriding adalah mendefinisikan kembali method yang sama, baik nama method maupun signature atau parameter yang diperlukan dalam subclass, kalau method overloading adalah mendefinisikan method yang memiliki nama yang sama, tetapi dengan signature yang berbeda dalam definisi class yang sama.

Latihan 1 :

```
abstract class Bentuk
{
    protected int panjang;
    protected int lebar;
    public String getBentuk()
    {
        return "Dasar";
    }

    public abstract int hitungLuas();
}

class BujurSangkar extends Bentuk
{

    public BujurSangkar(int panjang1, int lebar1)
    {
        this.panjang = panjang1;
        this.lebar = lebar1;
    }

    @Override
    public String getBentuk()
    {
        return "Bujur Sangkar";
    }

    @Override
    public int hitungLuas()
    {
        return panjang * lebar;
    }
}
class segitiga extends Bentuk
{

    public segitiga(int panjang2, int lebar2)
    {
        this.panjang = panjang2;
        this.lebar = lebar2;
    }
```

```
    @Override
    public String getBentuk()
    {
        return "Segitiga";
    }

    @Override
    public int hitungLuas()
    {
        return (panjang * lebar )/2;
    }
}
```

Latihan 2 :

```
public class Hewan
{
    private String tipe;
    public Hewan(String tipe)
    {
        this.tipe = tipe;
    }
    public String toString()
    {
        return "Ini adalah "+tipe;
    }
    public void bersuara()
    {

    }
}

public class Anjing extends Hewan
{
    private String nama;
    private String jenis;
    public Anjing(String nama)
    {
        super("Anjing");
        this.nama = nama;
        jenis = "tidak diketahui";
    }
    public Anjing(String nama, String jenis)
    {
        super("Anjing");
        this.nama = nama;
        this.jenis = jenis;
    }
    public void bersuara()
    {
        System.out.println("Guk guk guk");
    }
    public String toString()
    {
        return super.toString()+"\nIni adalah "+jenis+" bernama "+nama;
    }
}
```

```
public class Ayam extends Hewan
{
    private String nama;
    private String jenis;
    public Ayam(String nama)
    {
        super("Ayam");
        this.nama = nama;
        this.jenis = "Tidak diketahui";
    }
    public Ayam(String nama, String jenis)
    {
        super("Ayam");
        this.nama = nama;
        this.jenis = jenis;
    }
    public String toString()
    {
        return super.toString()+"\nIni adalah "+jenis+" bernama "+nama;
    }
    public void bersuara()
    {
        System.out.println("Kukuruyuk petok petok");
    }
}

public class Kucing extends Hewan
{
    private String nama;
    private String jenis;

    public Kucing(String tipe, String nama)
    {
        super("Kucing");
        this.nama = nama;
        this.jenis = "Tidak Diketahui";
    }

    public Kucing(String tipe, String nama, String jenis)
    {
        super("Kucing");
        this.nama = nama;
        this.jenis = jenis;
    }

    @Override
    public void bersuara()
    {
        System.out.println("Meong...Meong...");
    }

    @Override
    public String toString()
    {
        return super.toString()+"\nIni adalah "+jenis+" bernama "+nama;
    }
}
```

```
public class polymorfisme
{
    public static void main(String[] args)
    {
        Hewan[] hewannya = {new Anjing("Broni","Herder"),
                            new Kucing("Kiti","Persia"),
                            new Ayam("Toki", "Buras")};
        Hewan pilihHewan;
        Random pilih = new Random();
        for(int i = 0; i < 5; i++)
        {
            pilihHewan = hewannya[pilih.nextInt(hewannya.length)];
            System.out.println("\nPilihan Anda:\n" + pilihHewan);
            pilihHewan.bersuara();
        }
    }
}
```

Modul-8

Judul : **Exception Handling**

Teori : Bugs dan error dalam sebuah program sangat sering muncul meskipun program tersebut dibuat oleh programmer berkemampuan tinggi. Untuk menghindari pemborosan waktu pada proses error-checking, Java menyediakan mekanisme penanganan exception.
Exception adalah singkatan dari Exceptional Events. Kesalahan (errors) yang terjadi saat runtime, menyebabkan gangguan pada alur eksekusi program. Terdapat beberapa tipe error yang dapat muncul. Sebagai contoh adalah error pembagian 0, mengakses elemen di luar jangkauan sebuah array, input yang tidak benar dan membuka file yang tidak ada.
Beberapa contoh dari exception yang pernah kita jumpai pada materi sebelumnya adalah : ArrayIndexOutOfBounds, IOException.
Objek exception yang dihasilkan dapat kita manfaatkan untuk mengetahui lebih lanjut mengenai error atau exception yang terjadi. Exception merupakan subclass dari class Throwable yang mendefinisikan beberapa method yang juga diwarisi oleh exception. Tiga method yang penting adalah :

- getMessage()

  Method ini mengembalikan isi pesan untuk menggambarkan exception yang terjadi.
- printStackTrace()

  Method ini menampilkan pesan error dan stack trace ke standard error output stream yang biasanya merupakan konsol windows apabila program merupakan program konsol.
- printStackTrace(PrintStream s)

  Method ini menampilkan pesan error ke objek PrintStream yang dijadikan parameter. Apabila ingin menampilkan pesan ke konsol, kita dapat menggunakan System.out sebagai parameter.

Latihan 1. Menangkap exception
//sebelum ada exception

```
public class BagiNol
{
    public static void main(String[] args)
    {
        System.out.println("Hasil 5 / 0 : "+(5/0));
    }
}
```

// setelah dipasang blok try-catch

```
public class BagiNol
{
    public static void main(String[] args)
    {
        try
        {
            System.out.println("Hasil 5 / 0 : "+(5/0));
        }
        catch (Exception e)
        {
            System.out.println("Pesan Error : "+e);
        }
```

```
        System.out.println("Setelah Exception");
    }
}
```

Lat 2 Blok Try-catch-finally

```
public class MemoriException
{
    public static void main(String[] args)
    {
        int[] myArray = new  int[5];
        try
        {
                for(int i=0; i<=5; i++)
                {
                    myArray[i] = i;
                }
        }
        catch(Exception e)
        {
            System.err.println("Ada pesan kesalahan : "+e);
        }
        finally
        {
            System.out.println("Finally");
        }
        System.out.println("Hello World");
    }

}
```

Lat 3. Melempar Exception

```
public class ThrowingClass
{
    static void myMethod() throws ClassNotFoundException
    {
      throw new ClassNotFoundException("just a demo");
    }
}
public class ThrowsDemo
{
    public ThrowsDemo()
    {
    }
    public static void main(String args[])
    {
         try
         {
           ThrowingClass.myMethod();
         }
         catch (ClassNotFoundException e)
         {
           System.out.println(e);
         }
    }
}
```

Modul-9

Judul : **JDBC (Java Database Connectivity)**

Teori : Java Database Connectivity mempelajari bagaimana menghubungkan Program Java dengan Program Database. Adapun untuk menghubungkan java dengan database memerlukan sebuah connector.

Dalam pembahasan kali ini kita akan coba membahas bagaimana menghubungkan program java dengan program database MySQL dengan menggunakan connector yang disebut mysql-connector-java. Connector tersebut berupa library yang dapat di download secara free di situs MySQL atau situs Java sendiri

Untuk menguasai penggunaan JDBC sendiri, anda disyaratkan mampu menggunakan MySQL, dan memhami query-query yang ada.

Langkah – langkah dalam menggunakan JDBC

1. Letakkan mysql-connector di dalam direktori java\jre\lib\ext
2. Pada program java buatlah statement sebagai berikut :
   - `Class.forName("org.gjt.mm.mysql.Driver"); //deklarasikan nama driver`
   - `Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/nama_database","username","password");`
   - `Statement stmt = con.createStatement();`
   - `String sql = "";`
   - `stmt.executeUpdate(sql); // mengeksekusi query`
   - `stmt.close(); //menutup statement`
   - `con.close(); // menutup koneksi`

Lat 1. Menampilkan data

Buatlah sebuah database dengan nama “Buku” (tanpa tanda kutip)

```
drop database if exists buku;
create database buku;
use buku;

create table buku
(
  id_buku int(10) NOT NULL auto_increment,
  ISBN varchar(14),
  judul varchar(60),
  id_penerbit int(10),
  penulis varchar(60),
```

```
  primary key(id_buku)
);

CREATE TABLE penerbit
(
  id_penerbit int(10) NOT NULL auto_increment,
  nama varchar(40) default NULL,
  kota varchar(20) default NULL,
  PRIMARY KEY  (id_penerbit)
);
```

Buat code dalam java

```
import java.sql.*;

public class BookList
{
    public static void main(String[] args) {
        Connection con = null;
        Statement stmt = null;
        ResultSet rs = null;

        try {
             // loading driver MySql/J
            Class.forName("org.gjt.mm.mysql.Driver");

            // mendirikan koneksi database MySql
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/buku","root","admin");

            // statement SQL statik
            stmt = con.createStatement();

            // eksekusi SQL statement untuk tabel " buku"
            rs = stmt.executeQuery("SELECT id_buku, ISBN, judul, nama, penulis FROM
buku, penerbit WHERE buku.id_penerbit = penerbit.id_penerbit");

            // ambil data result -set
            ResultSetMetaData rsmd = rs.getMetaData();
            int nColumn = rsmd.getColumnCount();

            // tampilkan judul kolom
            for (int i = 1; i<= nColumn; i++)
            {
                System.out.print(rsmd.getColumnName(i) + " | ");
            }
            System.out.println("\n");

            // tampilkan hasil query berupa data result-set dari tabel
            while (rs.next())
            {
                 System.out.println(rs.getLong(1) + "\t" +
                        rs.getString(2) + ", \"" +
                        rs.getString(3) + "\", (" +
                        rs.getString(4) + "), [" +
                        rs.getString("penulis") + "]");
```

```
            }
        }
        catch(ClassNotFoundException e) {
            System.err.println("Error Loading Driver : "+ e.getMessage());
        }
        catch(SQLException e){
            System.err.println("Error SQL : " + e.getMessage());
        }
        finally{
            // lepaskan database & resource JDBC
            try {
                if (rs != null) rs.close();
            }catch (SQLException sqle) {}
            try{
                if (stmt != null) stmt.close();
            }catch (SQLException sqle) {}

            // lepaskan koneksi database
            try{
                if (con != null) con.close();
            }catch (SQLException sqle) {}
        }
    }
}
```

Modul-10

Judul : **GUI-1**

Teori : Java menyediakan banyak tool seperti Abstract Windowing Toolkit dan Swing untuk mengembangkan aplikasi GUI yang interaktif.

Java Foundation Class (JFC) merupakan sekumpulan class-class Java yang digunakan untuk mengembangkan perangkat lunak berbasis GUI (Graphical User Interface). Selain itu, JFC juga mempunyai class-class yang digunakan untuk menambahkan fungsi dan kemampuan interaksi yang variatif dari pemrograman Java. Dari definisi ini, JFC tidak hanya berisi class-class GUI saja tetapi juga class-class lain yang dapat meningkatkan kemampuan pemrograman Java baik dari segi fungsionalitasnya maupun dari segi kemampuan interaksi pemrograman Java yang sangat kaya.

AWT dan Swing menyediakan komponen GUI yang dapat digunakan dalam membuat aplikasi Java dan applet. Anda akan mempelajari applet pada bab berikutnya. Tidak seperti beberapa komponen AWT yang menggunakan native code, keseluruhan Swing ditulis menggunakan bahasa pemrograman Java. Swing menyediakan implementasi platform-independent dimana aplikasi yang dikembangkan dengan platform yang berbeda dapat memiliki tampilan yang sama. Begitu juga dengan AWT menjamin tampilan look and feel pada aplikasi yang dijalankan pada dua mesin yang berbeda menjadi terlihat sama. Swing API dibangun dari beberapa API yang mengimplementasikan beberapa jenis bagian dari AWT. Kesimpulannya, komponen AWT dapat digunakan bersama dengan komponen Swing.

Hirarki Swing

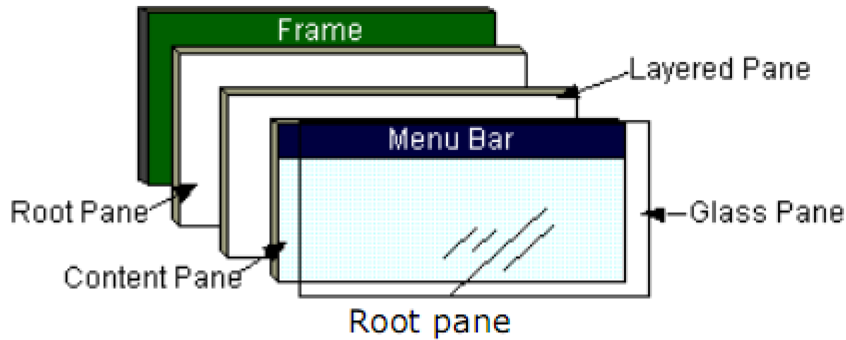
- **Top Level Containers**

  Komponen – komponen yang ada pada puncak setiap hirarki aplikasi swing
  Tampilannya bergantung pada platform dimana dia berada sehingga disebut heavyweight component
  Yaitu : Applet, Dialog, Frame

- **General Purpose Containers**
  Kontainer tingkat menengah yang dapat digunakan pada banyak situasi dan keadaan yang berbeda.
  Yaitu : panel, scroll pane, split pane, tabbed-pane, tool bar, split pane

- **Special-Purpose Containers**

  Kontainer antara yang mempunyai tugas khusus dalam User Interface.
  Yaitu : internal frame, layered pane



- **Basic Controls**

  Komponen-komponen yang berguna untuk memperoleh masukan dari pemakai, biasanya menunjukkan keadaan tertentu yang sederhana.
  Yaitu : Buttons, Combo Box, List, Menu, Slider, Text Fields
- **Uneditable Information Displays**

  Komponen-komponen yang berfungsi hanya untuk memberi informasi kepada pemakai.
  Yaitu : Label, Progress bar, Tool Tip
- **Editable Displays of Formatted Information**

  Komponen-komponen yang menampilkan informasi yang tersusun rapih dan bisa dipilih atau diedit oleh pemakai.
  Yaitu : Color Chooser, File Chooser, table, text, tree,

Lat 1 Menampilkan Frame Sederhana

```
import javax.swing.JFrame;
public class tampilFrame extends JFrame
{
    public static void main(String[] args)
    {
        tampilFrame frame = new tampilFrame();
        frame.setTitle("Testing Frame");
        frame.setSize(300, 300);
        frame.setVisible(true);
    }
}
```

Lat 2. Menampilkan Frame dengan Isi Label

```
import javax.swing.JFrame;
import javax.swing.JLabel;

public class helloWorld
{
    public static void main(String[] args)
    {
```

```
        JFrame frame = new JFrame("Hello World");
        JLabel label = new JLabel("Hello World");
        frame.getContentPane().add(label);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setSize(300, 400);
        frame.setLocation(150, 50);
        frame.setVisible(true);
    }
}
```

Lat 3. Menampilkan Button.

```
import java.awt.Container;
import javax.swing.*;

public class tampilPanel
{
    JFrame frame;
    JPanel panel;
    JTextField textField;
    JButton button;
    Container contentPane;

    public void launchFrame()
    {
        frame = new JFrame("My Aplikasi");
        panel = new JPanel();
        textField = new JTextField("Default text");
        button = new JButton("Klik Saya");
        contentPane = frame.getContentPane();
        panel.add(textField);
        panel.add(button);
        contentPane.add(panel);
        frame.pack();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
    public static void main(String[] args)
    {
        tampilPanel tmp = new tampilPanel();
        tmp.launchFrame();
    }
}
```

Modul-11

Judul : **GUI – 2 (Layout Manager)**

Teori : Posisi dan ukuran suatu komponen ditentukan oleh layout manager. Layout manager mengatur tampilan dari komponen di dalam kontainer. Berikut ini beberapa layout manager yang terdapat di dalam Java.

1. FlowLayout
2. BorderLayout
3. GridLayout
4. GridBagLayout
5. CardLayout

Layout manager dapat diatur menggunakan method setLayout dari class Container. Method ini dapat ditulis sebagai berikut.

```
void setLayout(LayoutManager mgr)
```

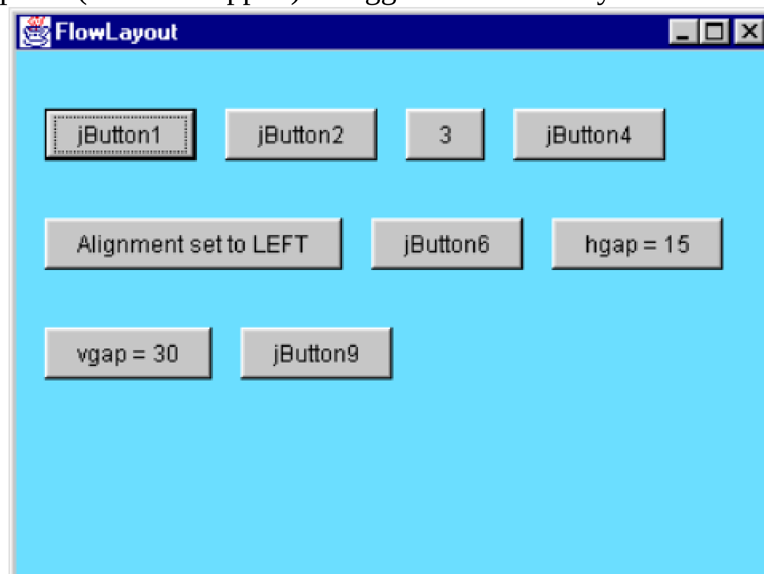
Jika Anda memilih untuk tidak menggunakan layout manager, Anda dapat mengisi null sebagai argumen untuk method ini. Tetapi selanjutnya, Anda akan mengatur posisi elemen secara manual dengan menggunakan method setBounds dari class Components.

```
public void setBounds(int x, int y, int width, int height)
```

Method ini mengatur posisi berdasarkan pada argumen x dan y, dan ukuran berdasarkan argumen width dan height. Hal ini akan cukup menyulitkan dan membosankan untuk aplikasi jika Anda memiliki beberapa objek komponen didalam object container. Anda akan memanggil method ini untuk setiap komponen.

1. Flow Layout

   FlowLayout menyusun komponen-komponen dalam baris-baris dari kiri ke kanan dan kemudian dari atas ke bawah menggunakan ukuran alami setiap komponen, preferredSize. FlowLayout membariskan sebanyak mungkin komponen yang dia mampu ke dalam satu baris, baru kemudian berpindah ke baris baru. Biasanya, FlowLayout digunakan untuk menyusun tombol-tombol pada sebuah panel. Dalam Java AWT, semua all panel (termasuk applet) menggunakan FlowLayout secara default.

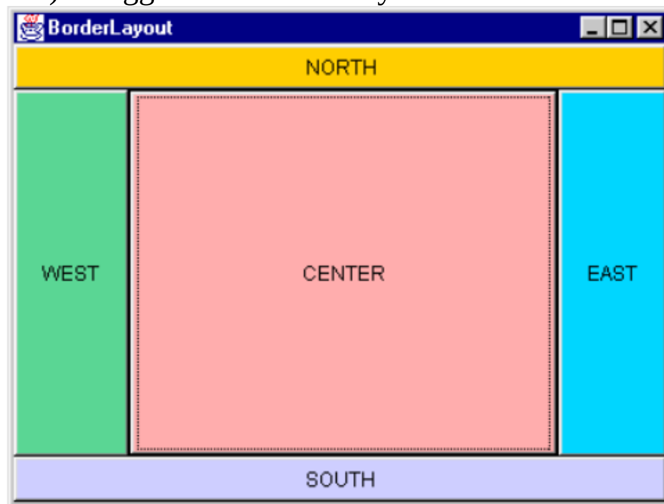


Gambar Flow Layout

Anda dapat memilih apakah komponen-komponen dalam satu baris itu tersusun rata-kiri, rata-kanan, atau berada di tengah-tengah baris dengan menset alignment justification dari FlowLayout left, right, atau center. Anda juga dapat menentukan besar pemisahan (horizontal dan vertical spacing) antar komponen komponen dan baris. Gunakan Component Inspector untuk mengubah kedua properti alignment dan gap ketika anda berada pada GUIEditing workspace.

2. Border Layout
   BorderLayout menyusun komponen-komponen dari sebuah kontainer dalam wilayah-wilayah yang dinamai North, South, East, West, dan Center.
   - Komponen-komponen di North dan South diberikan tinggi yang sesuai dan meregang sepanjang lebar seluruh kontainer.
   - Komponen-komponen di East dan West diberikan lebar yang sesuai dan meregang secara vertikal untuk memenuhi ruang antara wilayah North dan South.
   - Komponen-komponen yang terletak di Center mengembang untuk mengisi ruang yang tersisa.

   Dalam Java AWT (Abstract Window Toolkit), semua jendela (termasuk frame dan dialog box) menggunakan BorderLayout secara default.



Gambar Border Layout

   BorderLayout sangat baik untuk memaksa komponen-komponen menempel pada sisi-sisi kontainer dan untuk memenuhi bagian tengah dari kontainer dengan sebuah komponen.

3. Grid Layout
   Dengan GridLayout manager, komponen juga diposisikan dari kiri ke kanan dan dari atas ke bawah seperti pada FlowLayout manager. GridLayout manager membagi kontainer menjadi baris dan kolom. Semua region memiliki ukuran yang sama. Hal tersebut tidak mempedulikan ukuran sebenarnya dari komponen.

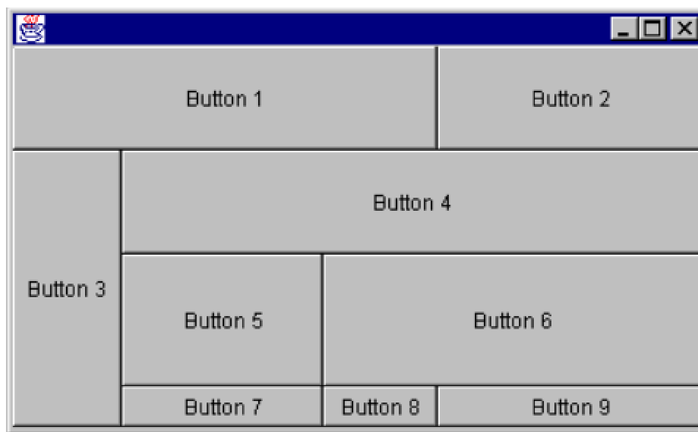


Gambar Grid Layout

Gunakan GridLayout jika anda merancang kontainer dimana komponen-komponennya mempunyai ukuran yang sama sepert misalnya tombol-tombol kalkulator.

4. GridBag Layout

   GridBagLayout adalah layout yang sangat luwes dan ampuh karena dengannya kita bisa mengatur dengan bebas tata letak komponen pada kotak-kotak sel dibandingkan dengan GridLayout. GridBagLayout menempatkan komponen-komponen secara horisontal dan vertikal pada kotak-kotak yang dinamis. Komponen-komponen tersebut tidak harus mempunyai ukuran yang sama, dan mereka dapat mengisi kotak lebih dari satu sel.



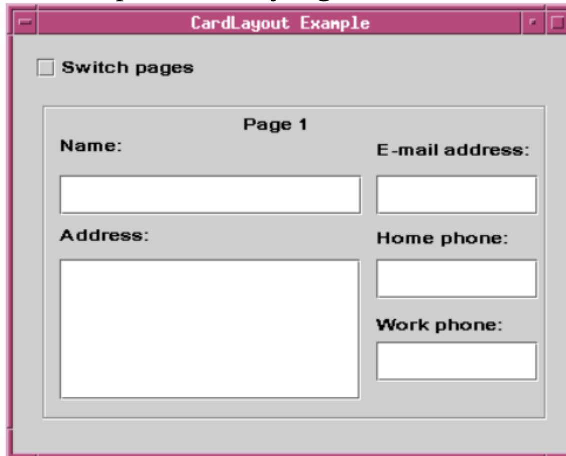
Gambar GridBag Layout

GridBagLayout menentukan penempatan komponen-komponennya berdasarkan batas-batas dan ukuran minimum dari setiap komponen, ditambah dengan ukuran alamikontainer.

Meskipun GridBagLayout dapat mengakomodasi grid yang kompleks, ia akan lebih baik (dan lebih mudah ditebak perilakunya) apabila kita mengatur komponen-komponen ke dalam panel-panel yang lebih kecil yang ditampung di dalam kontainer GridBagLayout. Panel-panel ini dapat menggunakan layout-layout yang lain atau bahkan mengandung panel-panel lagi. Metode ini mempunyai dua keuntungan:

- Kita dapat mengatur penempatan dan ukuran dari sebuah komponen dengan lebih natural karena menggunakan layout tertentu.
- Sel yang digunakan jadi lebih sedikit, dan membuat GridBagLayout menjadi lebih sederhana dan mudah dikontrol.

5. Card Layout
CardLayout menempatkan komponen-komponen (biasanya panel) satu di atas yang lainnya seperti tumpukan kartu. Kita hanya bisa melihat satu dalam satu waktu, dan anda dapat mengganti ke panel yang lain dengan menggunakan kontrol lain yang memilih panel mana yang berada di sebelah atas.



Gambar Card Layout

CardLayout merupakan layout sesuai untuk daerah yang mempunyai komponen berbeda pada saat yang berbeda pula. Hal ini membuat kita bisa mengatur dua atau lebih panel yang perlu menempati tempat yang sama.
CardLayout biasanya dihubungkan dengan komponen pengontrol seperti sebuah check box atau sebuah list. Keadaan dari komponen pengontrol menentukan komponen mana yang akan ditampilkan oleh CardLayout. Pengguna membuat pilihan dengan mengubah keadaan komponen pengontrol pada UI.

Lat 1. Membuat Flow Layout

```
import java.awt.*;
Import javax.swing.*;
public class flowLayout extends Frame
{
    public static void main(String[] args)
    {
        flowLayout demo = new flowLayout();
        demo.setTitle("Demo Layout");
        demo.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 10));
        demo.add(new JButton("satu"));
        demo.add(new JButton("Dua"));
        demo.add(new JButton("Tiga"));
        demo.setSize(100,100);
        demo.pack();
        demo.setVisible(true);
    }
}
```

Lat 2. Membuat Border Layout

```
import javax.swing.*;
import java.awt.*;
```

```
public class BorderLayout
{
    JFrame frame;
    JPanel panel;
    JButton btn1;
    JButton btn2;
    JButton btn3;
    JButton btn4;
    JButton btn5;
    Container content;
    BorderLayout layout;

    public void run()
    {
        frame = new JFrame("Border Layout Demo");
        panel = new JPanel();
        content = frame.getContentPane();
        layout = new BorderLayout();

        btn1 = new JButton("Button 1");
        btn2 = new JButton("Button 2");
        btn3 = new JButton("Button 3");
        btn4 = new JButton("Button 4");
        btn5 = new JButton("Button 5");

        panel.setLayout(new java.awt.BorderLayout());
        panel.add(btn1, java.awt.BorderLayout.SOUTH);
        panel.add(btn2, java.awt.BorderLayout.CENTER);
        panel.add(btn3, java.awt.BorderLayout.EAST);
        panel.add(btn4, java.awt.BorderLayout.WEST);
        panel.add(btn5, java.awt.BorderLayout.NORTH);

        content.add(panel);
        frame.pack();
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    public static void main(String[] args)
    {
        BorderLayout border = new BorderLayout();
        border.run();
    }
}
```

Modul-12

Judul : **GUI – 3 (Event Handling)**
Teori : Pada modul ini, kita akan belajar bagaimana mengendalikan events triggered ketika user berinteraksi dengan aplikasi GUI Anda. Setelah menyelesaikan modul ini, kita akan dapat mengembangkan aplikasi GUI yang dapat merespon interaksi user.

Delegation Event Model

Delegasi event model menguraikan bagaimana program Anda dapat merespon interaksi dari user. Untuk memahami model, pertama-tama mari kita pelajari melalui tiga komponen utamanya.

1. Event Source

Event source mengacu pada komponen GUI yang meng-generate event. Sebagai contoh, jika user menekan tombol, event source dalam hal ini adalah tombol.

2. Event Listener/Handler

Event listener menerima berita dari event-event dan proses-proses interaksi user. Ketika tombol ditekan, listener akan mengendalikan dengan menampilkan sebuah informasi yang berguna untuk user.

3. Event Object

Ketika sebuah event terjadi (misal, ketika user berinteraksi dengan komponen GUI), sebuah object event diciptakan. Object berisi semua informasi yang perlu tentang event yang telah terjadi. Informasi meliputi tipe dari event yang telah terjadi, seperti ketika mouse telah di-klik. Ada beberapa class event untuk kategori yang berbeda dari user action. Sebuah event object mempunyai tipe data mengenai salah satu dari class ini.

Lat 1. Membuat Demo JButton

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;

public class demoJButtonEvent
{
    JFrame frame;
    JPanel panel;
    JButton btn;
    JLabel label;
    Container content;

    public void run()
    {
        frame = new JFrame("Demo Event Handling");
        panel = new JPanel();
        btn = new JButton("Generate");
        btn.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                buttonAction(e);
            }
```

```
        });
        label = new JLabel("Random");
        label.setHorizontalAlignment(SwingConstants.CENTER);
        label.setFont(new Font("Tahoma",1,24));
        label.setBorder(BorderFactory.createBevelBorder(BevelBorder.RAISED));

        content = frame.getContentPane();

        panel.setLayout(new java.awt.BorderLayout());
        panel.add(btn, BorderLayout.SOUTH);
        panel.add(label, BorderLayout.CENTER);

        content.add(panel);
        frame.pack();
        frame.setSize(300, 300);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }
    public void buttonAction(ActionEvent e)
    {
        int random = (int) Math.floor(Math.random() * 10000);
        label.setText("["+random+"]");
    }
    public static void main(String[] args)
    {
        demoJButtonEvent btnDemo = new demoJButtonEvent();
        btnDemo.run();
    }
}
```

Lat 2 Membuat demo file chooser

```
import java.awt.*;
import java.awt.event.*;
import java.io.File;
import javax.swing.*;
import javax.swing.event.*;

public class demoFileChooser extends JPanel implements ActionListener
{
    static private final String newline = "\n";
    JButton openButton, saveButton;
    JTextArea textArea;
    JFileChooser fileChooser;

    public demoFileChooser()
    {
        super(new BorderLayout());
        //buat textArea dahulu, karena action listener mengacu pada itu
        textArea = new JTextArea(5,20);
        textArea.setMargin(new Insets(5,5,5,5));
        textArea.setEditable(false);
        JScrollPane logScrollPane = new JScrollPane(textArea);

        // buat JFileChooser
        fileChooser = new JFileChooser();
```

```
        //fc.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
        fileChooser.setFileSelectionMode(JFileChooser.FILES_AND_DIRECTORIES);
        openButton = new JButton("Open File....",createImageIcon("open16.gif"));
        openButton.addActionListener(this);

        saveButton = new JButton("Save File...",createImageIcon("save16.gif"));
        saveButton.addActionListener(this);

        JPanel buttonPanel = new JPanel();
        buttonPanel.add(openButton);
        buttonPanel.add(saveButton);

        add(buttonPanel, BorderLayout.PAGE_START);
        add(logScrollPane, BorderLayout.CENTER);
    }

    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource() == openButton)
        {
            int returnVal = fileChooser.showOpenDialog(demoFileChooser.this);
            if(returnVal == JFileChooser.APPROVE_OPTION)
            {
                File file = fileChooser.getSelectedFile();
                textArea.append("Opening : "+file.getName()+"."+newline);
            }
            else
            {
                textArea.append("Open Command Canceled by User."+newline);
            }
            textArea.setCaretPosition(textArea.getDocument().getLength());
        }
        else if(e.getSource() == saveButton)
        {
            int returnVal = fileChooser.showOpenDialog(demoFileChooser.this);
            if(returnVal == JFileChooser.APPROVE_OPTION)
            {
                File file = fileChooser.getSelectedFile();
                textArea.append("Saving : "+file.getName()+"."+newline);
            }
            else
            {
                textArea.append("Save Command canceled by User. "+newline);
            }
            textArea.setCaretPosition(textArea.getDocument().getLength());
        }
    }

    protected static ImageIcon createImageIcon(String path)
    {
        java.net.URL imageURL = demoFileChooser.class.getResource(path);
        if(imageURL != null)
        {
            return new ImageIcon(imageURL);
        }
        else
        {
            System.err.println("Tidak dapat menemukan file "+path);
```

```
            return null;
        }
    }

    private static void createAndShowGUI()
    {
        JFrame frame = new JFrame("File Chooser Demo");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.add(new demoFileChooser());
        frame.pack();
        frame.setVisible(true);
    }
    public static void main(String[] args)
    {
        SwingUtilities.invokeLater(new Runnable()
        {
            public void run()
            {
                UIManager.put("swing.boldMetal", Boolean.FALSE);
                createAndShowGUI();
            }
        });
    }
}
```

Lat 3 Membuat demo Combo box

```
import java.awt.*;
import javax.swing.*;
import java.net.*;
import java.awt.event.*;
/**
 *
 * @author Didan
 * Universitas Muhammadiyah Cirebon
 */
public class demoComboBox extends JFrame
{
    JLabel labelPic;
    JComboBox cb;
    private void run()
    {
        labelPic = new JLabel();
        cb = new JComboBox();

        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setTitle("Demo Combo Box");

        labelPic.setHorizontalAlignment(SwingConstants.CENTER);
        labelPic.setIcon(new ImageIcon(getClass().getResource("/duke.jpg")));

        cb.setModel(new DefaultComboBoxModel(new String[]
{"duke","butterfly","eagle","pegasus","horse"}));
        cb.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
```

```
            {
                comboBoxAction(e);
            }
        } );

        getContentPane().add(cb, BorderLayout.PAGE_START);
        getContentPane().add(labelPic,BorderLayout.CENTER);

        pack();
        setVisible(true);
        setSize(300,300);

    }
    private void comboBoxAction(ActionEvent evt)
    {
        cb = (JComboBox) evt.getSource();
        String str = (String) cb.getSelectedItem();
        URL url = getClass().getResource("/"+str.toLowerCase()+".jpg");
        ImageIcon icon;
        if(url != null)
        {
            icon = new ImageIcon(url);
        }
        else
        {
            System.err.println("Image Not Found : /"+str+".jpg");
            return ;
        }
        labelPic.setIcon(icon);
    }
    public static void main(String[] args)
    {
        demoComboBox combo = new demoComboBox();
        combo.run();
    }

}
```

Modul-12

Judul : **Applet**

Teori : Applet adalah satu fitur yang paling menarik dalam java. Applet merupakan program yang Anda jalankan melalui web browser. Pada pembelajaran ini Anda akan belajar tentang membuat sebuah applet.

Sebuah applet adalah tipe yang spesial dari program java yang dieksekusi melalui internet. Secara khusus berjalan pada suatu web browser seperti Netscape Navigator, Mozilla, atau Microsoft Internet Explorer. Bagaimanapun juga, jika dibandingkan dengan aplikasi Java yang normal, tidak diperbolehkan mengakses applet pada komputer yang dijalankan untuk alasan keamanan. Applet ini cukup terbatas jika dibandingkan dengan aplikasi Java.

Method-method dalam Applet

a. Siklus applet

Bahkan untuk memulai eksekusi pada method main seperti dalam aplikasi khas Java, browser atau applet viewer berhubungan dengan applet melalui method-method berikut:

1. init()

init adalah method yang dipanggil pertama kali. Yang sebenarnya berisi permintaan pertama ketika applet di load.

2. start()

Setelah meminta method init, mulai dengan method yang dipangil selanjutnya. method ini meminta dokumen HTML yang ditampilkan applet setiap waktu. Eksekusi ringkasan dengan method ini dilakukan ketika applet ditampilkan kembali.

3. stop()

Ketika web browser meninggalkan dokumen HTML applet,method ini dipanggil untuk menginformasikan applet bahwa dia harus menghentikan proses eksekusinya.

4. destroy()

Method ini dipanggil ketika applet perlu dihapus dari memory. Method stop selalu dipanggil sebelum method ini diminta untuk dijalankan.

b. Paint

Method lain yang tidak kalah penting adalah method paint, yang mana class Applet menurunkannya dari class induknya yaitu class Component, yang meminta output applet setiap waktu yang diperlukan untuk dapat digambar kembali. Sebagai contoh dari setiap instance adalah ketika sebuah applet tersembunyi oleh window lain dapat dibuat terlihat lagi. Method ini selalu menolak ketika Anda ingin membuat bagaimana applet yang Anda buat harus terlihat seperti yag Anda inginkan. Pada contoh Hello World, applet memiliki string "Hello world!"pada background setelah menolak method paint.

c. Method Show Status

Applet memiliki window status, dimana memberi informasi kepada Anda tentang apa yang sebenarnya dilakukan applet. Jika Anda ingin memberi output ke window status, secara sederhana memanggil method showStatus.

Lat 1. Membuat Applet Sederhana

```
import java.awt.*;
import java.applet.*;

public class appletHello extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("Hello World", 80, 25);
    }
}
```

Lat 2 Membuat Applet huruf bergerak

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class simpleApplet extends Applet
{
    String teks;
    int warna;
    Font huruf = new Font("Tahoma", Font.BOLD, 40);

    private int counter = 0;
    private int base = 0;
    private int direction = 0;

    Color backgroundColor = new Color(250,0,255);

    public void init()
    {
        try
        {
            teks = getParameter("teks");
            warna = Integer.parseInt(getParameter("color"),16);
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
        if(teks == null)
        {
            teks = "Selamat Datang";
        }
        setBackground(backgroundColor);
    }

    public void paint(Graphics g)
    {
        // set huruf
        g.setFont(huruf);

        //ambil ukurn font
        FontMetrics fm = g.getFontMetrics();

        //ambil ukuran panjang dan tinggi string
```

```
        int panjang = fm.stringWidth(teks);
        int tinggi = fm.getHeight();

        //hitung posisi kiri
        int kiri = (this.getWidth() - panjang) /2;

        //hapus teks sebelumnya
        g.setColor(backgroundColor);
        g.drawString(teks, kiri, base);

        //hitung posisi base (tengah vertikal) yang baru
        if(counter == 0)
        {
            base = (this.getHeight() + tinggi) / 2;
        }
        else
        {
            //perbarui conter
            if((base - tinggi / 2) <= 0)
                direction = 1;
            else if (base >= this.getHeight())
                direction = 0;

            if(direction == 0)
                base--;
            else
                base++;
        }
        //tampilkan di tengah applet
        g.setColor(new Color(warna));
        g.drawString(teks, kiri, base);

        //delay
        for(int i = 0; i < 1000000; i++);
        counter++;
        repaint();

    }

    public void update(Graphics g)
    {
        if(isShowing())
        {
            paint(g);
        }
    }
}
```

Lat 3 Membuat demo life cycle applet

```
import java.applet.*;
import java.awt.*;

public class LifeCycleDemo extends Applet
{
    String msg = "";
```

```
    public void init()
    {
        msg += "Inisialisasi....";
        repaint();
    }
    public void start()
    {
        msg += "Starting....";
        repaint();
    }
    public void stop()
    {
        msg += "Stopping...";
        repaint();
    }
    public void destroy()
    {
        msg += "Prepare fo unloading... ";
        repaint();
    }
    public void paint(Graphics g)
    {
        g.drawString(msg, 15, 100);
    }
}
```